September 1993 ADODCAS 1993

INCREASING MODEL EFFICIENCY USING STANDARD COMMERCIAL SOFTWARE

Mr. Joel Ditto (205) 876-3290 &
Mr. David Morrison (205) 842-1142

SFAE-ASM-LS-BM-C
Redstone Arsenal AL 35898-8051

Annual Department of Defense Cost Analysis Symposium Paper

See Over

Software Model
Commercial

U	U	U	UL

INCREASING MODEL EFFICIENCY USING STANDARD COMMERCIAL SOFTWARE

This paper will demonstrate how the U.S. Army Missile Command (MICOM) Cost Model has been made more efficient using up-to-date, commercial software, specifically Windows and EXCEL.

The MICOM Cost Model does an excellent job of number crunching, has built in capabilities to produce the required (Army Cost Matrix) reports and can generate automated documentation (Cost Data Sheets and Variable Explanation Sheets). These capabilities are still at the forefront of cost model capabilities, but the user interface is not as up to date since the model is written in FORTRAN. We chose to use EXCEL in the Windows environment in order to make data entry more efficient and simplify creation of Ad Hoc Reports; while continuing to use the MICOM Cost Model calculation module to do actual cost calculations. The enhanced editing capability and the ability to view multiple open files makes model use much more efficient and greatly reduces the time required to perform quick turn-around cost excursions. Even though we used EXCEL, several other commercial software packages (i.e., LOTUS 1-2-3, Quattro Pro, etc.) could have been used instead.

Mr. Joel Ditto
(205) 876-3290
Mr. David Morrison
(205) 842-1142 or DSN 788-1142
SFAE-ASM-LS-BM-C
Redstone Arsenal AL 35898-8051

(3)

Increasing Model Efficiency Using Standard Commercial Software
Automated Cost Models Workshop

A. INTRODUCTION

1. Current Models Developed or Used by the U.S. Army
2. LOSAT Model Choice (MICOM Cost Model)
3. Problems With Existing Cost Models and In particular with the MICOM Cost Model (PICES)

B. HOW PICES WORKS

1. File Types and Structure
2. Program Modules

C. COMPARISON OF OLD AND NEW DATA ENTRY METHODS

1. Variable Factor File
2. Deployment Schedule File
3. Cost Account File

D. SAMPLE CASE DEMONSTRATION

1. Data file Change Example
2. Ad-Hoc Reports Example
3. Automated Documentation Retrieval Example

E. CONCLUSIONS AND RECOMMENDATIONS

Increasing Model Efficiency Using Standard Commercial Software

A. INTRODUCTION - The purpose of this workshop is to demonstrate how we use commercial, Windows-based software to more efficiently use an existing government program which produces Baseline Cost Estimates (BCEs). In addition we will make some suggestions for government software development projects.

1. Current models developed or used by the U.S. Army to produce Baseline Cost Estimates (BCEs) include ACEIT, OBCE, PICES and FLEX. Spreadsheets are also used to create a large number of BCEs. Each model has its strengths and weaknesses, based on choices made during its development concerning the language to be used, platform on which to run, etc. All the models mentioned have character based user interfaces.

2. At the Line of Sight Anti-Tank (LOSAT) Project Office we currently use the MICOM Cost Model, PICES, which in the past has been used to prepare about 50% of the BCEs done in the Army for major weapon systems.

3. Like other models, PICES has its strong and weak points. PICES is written in FORTRAN and thus can be compiled on many platforms. The primary computer system used today is the (DOS based) Personal Computer (PC). PICES calculates large estimates quickly, has built in capabilities to produce required (Matrix type) reports and can generate automated Cost Data Sheets and Variable Explanation Sheets. The choice of FORTRAN gives PICES high speed number crunching, but its poor Input/Output (I/O) capabilities make it extremely difficult to create a modern user interface. The PICES data edit program makes maximum use of FORTRAN's limited I/O capabilities, but it is still difficult to enter large quantities of data and there are no capabilities for search and replace, global replace, etc. PICES produces the required standard reports, but simpler Ad Hoc reports are needed. We chose to attack these I/O problems by using commercial Windows-based software (specifically EXCEL) to create and edit input files and to quickly produce formatted output.

B. HOW PICES WORKS

1. PICES consists of a number of programs designed to run from a database of seven input data files. The information in these files is the same as that which would be required for any cost estimate, formatted as required by the various program modules. The data files are described below.

Report Generation File - Contains the basic descriptive information required, such as program name, years to calculate, sunk years, etc.

Variable Factor File - Contains single valued variables such as first unit cost, learning curve, etc..

Deployment Schedule File - Contains variables which are changed on a yearly or monthly basis. Some examples would be annual production quantities, monthly deliveries, manyears of effort, throughputs of known contracts, etc. . The data consists of a start year, then the values the variable assumes for each month or year.

Time Phasing File - Contains variables similar to deployment schedules, but the start time for the variable is contained in a separate, milestone file.

Milestone File - Contains Milestones which specify the starting point for Time Phasing Variables.

Inflation/Escalation Factor File - Contains factors to change costs from one base year dollars to another, and escalation factors to calculate escalated costs from constant dollar inputs.

Cost Account File - Contains the various cost accounts into which the estimate is divided. Each cost account contains the data which describes how the cost for that account is calculated, i.e. the equation, how the costs are spread, etc.

In addition to the input data files, PICES creates output files and report files.

2. As stated previously, PICES has several program modules that run from the files mentioned above. All the program modules are written in FORTRAN77. These programs can be accessed from a menu, or can be run directly, and are described below.

DATA FILE CREATION AND EDIT PROGRAM - Provides capability to enter or edit data for the seven input files. The editing capabilities and user interface are limited by the capabilities of FORTRAN.

HARDCOPY GENERATION PROGRAM - Prints a hard copy of any of the seven input data files.

COST ACCOUNT FILE MERGE PROGRAM - Provides capability to put two cost account files together. This essentially allows you to "add" two separate cost estimates.

COST CALCULATION PROGRAM - This module does the actual cost calculations, allows the analyst to view results of previous calculations, and stores results in output files.

COST ACCOUNT FILE RE-CREATION PROGRAM - Provides some recovery capabilities for corrupted files.

COST UNCERTAINTY ANALYSIS PROGRAM - Provides a framework for quantifying uncertainty. Costs for an account can be calculated according to a Uniform, Beta or Triangular distribution.

INPUT FILE TRANSLATION/COMPRESSION PROGRAM - Translates the seven input files from their usual binary form to ASCII or CSV (Comma Separated Variable) format and back. The capability to translate from ASCII or CSV format to binary is also embedded in the calculation program.

COST ACCOUNT RENUMBER PROGRAM - Provides the capability to renumber (reference numbers) the Cost Account File..

COST OUTPUT FILE PRINT PROGRAM - Prints output files in P92 format or the new Army Cost Handbook formats.

RATIONALE AND METHODOLOGY PRINT PROGRAM - Creates Cost Data Sheets and Variable explanation sheets in several formats (P-92 or Cost Handbook).

SPREADSHEET INTERFACE PROGRAM - Translates cost output files from PICES format to ASCII or CSV format. This capability is also embedded in the calculation program.

MATRIX COST ACCOUNT FILE CREATION PROGRAM - Automatically generates a WBS Matrix oriented (as opposed to time phased) cost account file from the basic cost account file.

C. COMPARISON OF OLD AND NEW DATA ENTRY METHODS

The "normal" way to use PICES would be to create and edit the data files required using the data file creation and edit program, calculate costs using the cost calculation program, then use the other programs as required to do further analyses and produce the required reports and documentation for the BCE. With PICES used in this fashion, the data file creation and edit program is by a tremendous margin the most used program, yet it reflects the weakest part of FORTRAN - interactive data entry. In the following demonstration, the "normal" data editing procedures will be shown, along with greatly improved methods using commercial software.

All of the input files necessary to run the calculation module of PICES can be built with some kind of text editor, instead of the Data File Creation and Edit Module. This allows the user to develop the cost model with the editing tools most familiar to him. Examples of programs with good text editing capabilities are Windows Notepad, the VI editor in UNIX, spreadsheets such as EXCEL and LOTUS 123, word processors like WordPerfect or Word for Windows, and databases, such as dBASE, ACCESS or PARADOX. Any of these commercial programs have editing capabilities far exceeding

those of PICES or the other BCE models mentioned previously.

We chose to use EXCEL to edit and create our files, with the ASCII comma separated variable (CSV) file type . The edit functions of EXCEL and the Windows graphic interface works well for us and allows us to edit and create files using all of the power of the spreadsheet interface. PICES can translate both CSV and standard ASCII files to the binary files it uses for calculation, and can translate output files to CSV format as well.

Though there are seven input files, only three are updated and changed with any frequency. These are the Deployment Schedule, Variable Factor and the Cost Account files. We will show how these files can be created/edited with either the PICES editor or in EXCEL. We will be running standard PICES on one screen and using EXCEL on another. We assume that our audience is familiar either with EXCEL or some other spreadsheet.

1. Variable Factor File - ***Show how the Variable Factor File is edited in both PICES and EXCEL. In EXCEL, show how the screen is formatted, and when it is time to move on to the next file type, size the window and move it to the proper screen location for later use.***

Variable factor file notes: You can have up to 2015 variable factors, our BCE currently has between 450 and 500. With PICES you can go directly to a variable factor only if you know its variable factor number (i.e. address), otherwise you must go through them about 15 at a time intil you find the one you want. In EXCEL you can search for a variable factor number, search for a name if you know part of the name, or even just search for the value if you don't know the name or number. You can also split the screen if you need to look at factors located far apart and look at both factors at the same time. When using CSV files, EXCEL stores only values (not formulas) when the spreadsheet is saved. Therefore you can calculate the value needed for a variable, then when you save the file only the value will be there for PICES to use in calculations.

Deployment Schedule File - ***Show how the Deployment Schedule File is edited in both PICES and EXCEL. When it is time to move on to the next file type, size the window and move it to the proper screen location for later use.***

Deployment Schedule (DP) File notes: Monthly schedules are numbered 1-98 and annual schedules are numbered 101-500. With PICES you specify which deployment schedule you want to edit, then the second screen lets you modify the DP name, the third screen lets you modify the DP start year and the fourth screen lets you edit the monthly or yearly values. In EXCEL, you simply edit the DP, which consists of 1-5 rows, depending on how many years have data. The same superior search and screen functions are available as previously described.

3. Cost Account File - The last file we will discuss in detail is the Cost Account file. It is the most difficult to work with because its format changes with the different types of cost accounts. The format of the record depends on the type of Cost Estimating Relationship (CER) used in the account. Here a CER is merely a type of calculation, not a statistically derived relationship. Examples of CER types include throughputs, equations, and learning curves. ***Show how the Cost Account File is edited in both PICES and EXCEL. When the EXCEL description is complete, size the window so that all three file types are now on the screen.***

One of the most important advantages of editing files using EXCEL is the capability to see several files on the screen at the same time. As we have opened CSV PICES files, we have not bothered to close any previously open file(s). All the files we have opened are still open and visible at the same time. The PICES edit program allows you to have several files open, but only one at a time can be viewed.

D. SAMPLE CASE DEMONSTRATION

1. In order to demonstrate the enhanced capabilities of editing input files using EXCEL instead of the PICES editor we will make some changes in the sample case already shown. In our sample case, initial spares were initially calculated as a percent of total manufacturing cost. We have since decided to estimate initial spares by the major categories of Missile, Warhead and Launcher. Government logistics personnel have told us that initial spares is a function of recurring manufacturing cost, 50% the first year of production, 25% the second year, and 7% for succeeding years. This will require the creation of three new cost accounts and changing the CER type of the current account. ***Do the changes in PICES and EXCEL. Also show how output files can be viewed in both PICES and EXCEL.***

2. Once costs have been calculated, they can be saved in ASCII or CSV format, transported across platforms, or manipulated as desired. The files can be loaded into a spreadsheet where additional computational work could be performed on these files such as restructures, cost comparisons to other cost runs, budget drills, etc. These same files could also be input into a database program where they could be manipulated into various reports or views of the data. We will demonstrate the creation of an Ad Hoc report using a saved output file. Since our example consists of Procurement Dollar Costs only (Big 6 Cost Element 2.0), we have decided to make a report which summarized costs at the first level of indenture under procurement, i.e. 2.1, 2.2, etc. This same type of report could be done in PICES, but is much easier in a spreadsheet. ***Do the Ad Hoc report in EXCEL, and show how the same Ad Hoc spreadsheet could be re-used if we were doing a number of what-ifs.***

3. As mentioned previously, PICES has the capability to prepare the standard Cost Document Format (Cost Data Sheet) and Variable Explanation Format (Variable Data Sheet) suggested by the Army Cost Analysis Manual. The input for this documentation can be done with any editor, as long as the data is stored as standard

ASCII text. A number of (slightly) different formats have been used or suggested over the years including the (two page) documentation with both Cost Data Sheets and Variable Explanation Sheets, and (one page) documentation with a merged Cost Data Sheet/Variable Explanation Sheet format. Regardless of the format, this documentation requires a tremendous amount of paper to produce but provides no automated capabilities to the validator, the Cost Review Board (CRB) or the Cost Analysis Improvement Group (CAIG). We might also mention that to our knowledge, neither do any of the other BCE models mentioned.

Several Windows-based tools are available which could be used to eliminate the paper documentation and allow the reviewer to have documentation on line and keyed to calculated cost outputs. We have done a little work in this area; but only enough to show potential. Our approach was to save the one-page type documentation for our sample case as a Microsoft Word document, then use the OLE capabilities of Windows, EXCEL and Word to link the cost documentation to the Time Phased matrix of calculated costs. Other Windows-based applications (such as a database) could be used instead of the two mentioned. ***Do the DEMO of Output Keyed to Documentation. Note how the reviewer/validator's task is made easier when he can pop up documentation by double clicking (with a mouse) on the the reference number of an account.***

E. CONCLUSIONS AND RECOMMENDATIONS

As stated at the beginning of this presentation, every software package has its strong and weak points. We have discussed some weaknesses in PICES and how we have worked around them. We found PICES' problems to be in the area of data entry and Ad Hoc reporting. PICES, and the other BCE models initially mentioned are all hampered by a character based user-interface. We have also noted some government software is a special purpose version of a type of software available commercially in a general purpose form. This leads to a product with an inferior user interface and or feature set. An example might be writing a special purpose data base instead of writing a database application in any of the excellent commercial database products available. Sometimes software is written which is dependent on specific hardware or a specific release of some software product. This leads to expensive updates or in some cases, simply scrapping hardware and software.

All software is designed to perform some core application. In addition there are usually a number of other functions, such as I/O, report writing, graphics, etc., which must be performed but are not unique to the application. Nevertheless, these peripheral functions are programmed time and again for each new application and for each new hardware/software platform used in the workplace. In times of declining budget, the government cannot afford to recode peripheral function software for every application, or to produce software whose basic function is the same as that of standard commercial software products. It makes more sense to spend our efforts in developing the minimum amount of code needed to do the specialized functions and

leave the I/O to commercial software programs which do a much better job of performing those functions.

The following table lists the lines of source code for most of the programs in PICES. Also shown are the programs required for the core function (BCE production) of PICES. The other programs could be performed with commercial software.

Program	All Functions	Core Functions
DATA FILE CREATION AND EDIT PROGRAM	22,257	
HARDCOPY GENERATION PROGRAM	2,886	
COST ACCOUNT FILE MERGE PROGRAM	1,192	
COST CALCULATION PROGRAM	*13,106*	*13,106*
COST ACCOUNT FILE RE-CREATION PROGRAM	667	
COST UNCERTAINTY ANALYSIS PROGRAM	*6,829*	*6,839*
INPUT FILE TRANSLATION/COMPRESSION PROGRAM	6,009	
RENUMBER PROGRAM	*1,719*	*1,719*
COST OUTPUT FILE PRINT PROGRAM	*4,453*	*4,453*
RATIONALE AND METHODOLOGY PRINT PROGRAM	*10,235*	*10,235*
SPREADSHEET INTERFACE PROGRAM.	3,263	
MATRIX COST ACCOUNT FILE CREATION PROGRAM	725	
TOTAL	73,341	*36,342*

Only about half the lines of PICES code are necessary if commercial software is used for peripheral functions. It would also be possible to perform the functions of the Rationale and Methodology Print Program with commercial software. If that were done about 36% of the lines of PICES code are absolutely necessary. We believe about 50% of the lines of code normally written in a development effort could be eliminated by using commercial software to perform common functions. A significant amount of effort would of course be required to ensure that the core application modules interfaced with commercial software. Even if half the code savings were eaten up in this fashion, the development would be significantly less expensive and the basic application would be much less prone to obsolescence, since many functions would be performed by commercial software which is updated on a regular basis. We cannot afford the resources to recode common functions which can be performed by any number of commercial software packages.

Commercial software is headed in the same direction we are recommending for the government. Companies use the same graphics tools in their word processors and spreadsheets. Developers of all kinds of text editing software make their applications compatible with mail programs instead of developing a new mail program.

We hope you have benefitted from seeing how commercial software can be used to improve usage of current government software. Even more important, we hope that in the future, government software development is improved. Development cost and time could be cut significantly if the thrust of development began with determining the specialized functions needed, and then coding only those core functions; using commercial software to perform peripheral functions.